

IBM i World 2023

IBM i コンテンツ (2023年3月版)

IBM i で Python を使ってみよう

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Powerテクニカルセールス
澤田英寿

目次

IBM iでのPythonご使用をお勧めする理由

1. Python 概説/特徴
2. Pythonのインストール手順
 - 1) sshdの開始
 - 2) yumの導入
 - 3) Pythonの導入と確認
3. Pythonで簡単なプログラムを作ってみよう
 - 1) 既存のデータベースにアクセスしてみよう
 - 2) itoolkitを使ってIBM i コマンドにアクセスしてみよう
4. PythonでWebフレームワークを使ってみよう
 - 1) シンプルなWebフレームワーク「Bottle」のご紹介
 - 2) Bottleの導入と稼働テスト
 - 3) BottleのWebフレームワークを利用したWebアプリ
5. 補足情報

IBM iでのPythonご使用をお勧めする理由

- ✓ **ユーザー体験の向上**やデータの取り扱いに強い、汎用性の高い開発言語です。RPGやCOBOLでは苦手な、グラフィカルなアプリケーション開発、Webアプリ開発、**データ解析**、などは、いずれもPythonの得意分野です。
- ✓ Pythonのコーディングは**簡単**です。（他と比べてシンプルで学習し易い言語）。また、CLプログラム言語のように、可読性が高い言語です。
- ✓ コミュニティー（ボランティア団体）が多数の汎用性の高いライブラリーを開発しています。一から開発する必要がありません。
- ✓ Pythonは最も急成長している言語で、人気ランキングのTOPに常にランクインしています。**IBM i アプリ開発の人材確保**が容易になります。

1. Python 概説/特徴 (1/2)



- Python(パイソン)は1991年にオランダ人のグイド・ヴァン・ロッサム氏によって開発されたプログラミング言語です。1994年にv1.0がリリース。
- システム管理やツール・アプリケーション開発・科学技術計算・Webシステムなどで広く利用されている。特に、2010年代ごろからの機械学習ブームでは、優れた科学技術計算ツールとして使われてきた。
- インタープリター、マルチプラットフォームでコンパイル不要の言語
- 非営利団体であるPythonソフトウェア財団が管理しており、オープンソースとして提供されている

1. Python 概説/特徴 (2/2)



- 多数の拡張ライブラリー：PyPI（Python Package Index）が使える。
 - ー <https://pypi.org/> に現時点(3/6)で、43万6千のプロジェクトがありコミュニティに共有されている。
- 2000年10月にPython2.0リリース。2008年12月に、Python3.0リリース
- 2022年10月出荷のPython 3.11が最新バージョン

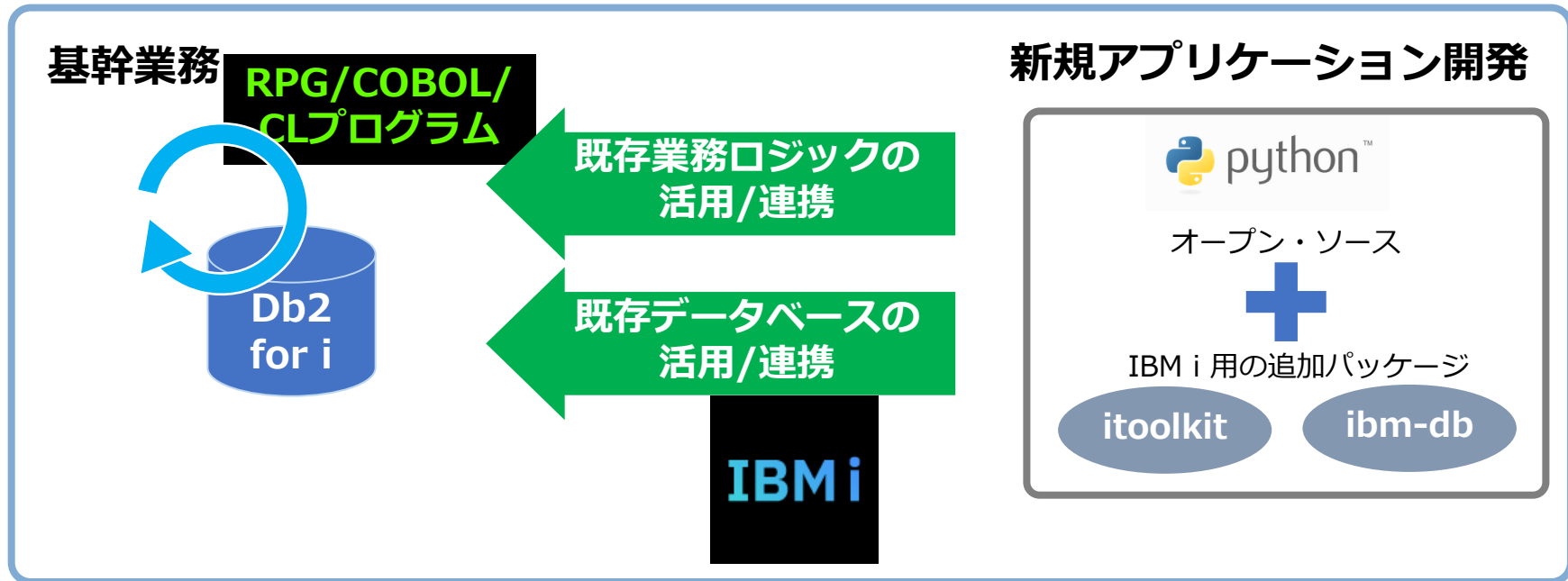
Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

既存アプリケーション資産とPython連携

- 1) PGMをはじめとしたシステム資源にアクセスするための **itoolkit**
 - 2) 基幹データ (Db2 for i) を操作するための **ibm_db**
- が、**IBM**のオープンソース・プロジェクトとして提供されている。



(参考) 既存アプリケーション資産とPython連携

1) PGMをはじめとしたシステム資源にアクセスするための**itoolkit**

本資料では、「**itoolkit**」を使った簡単なプログラムをご紹介します。

itoolkitは、IBM i のデータベース以外の IBM i システム資源へのアクセスを提供

- ・ コマンド
- ・ プログラム
- ・ システム状況
- ・ ジョブ情報
- ・ ユーザー空間
- ・ データ待ち行列
- ・ オブジェクト情報
- ・ . . .

詳細は、下記を参照してください。

<https://github.com/IBM/python-itoolkit>

2) 基幹データ (Db2 for i) を操作するための**ibm_db**

本資料では、「**ibm_db**」を使った簡単なプログラムをご紹介します。

ibm_dbは、DB2 for i データベース・オブジェクトへのアクセスを提供

詳細は、下記を参照してください。

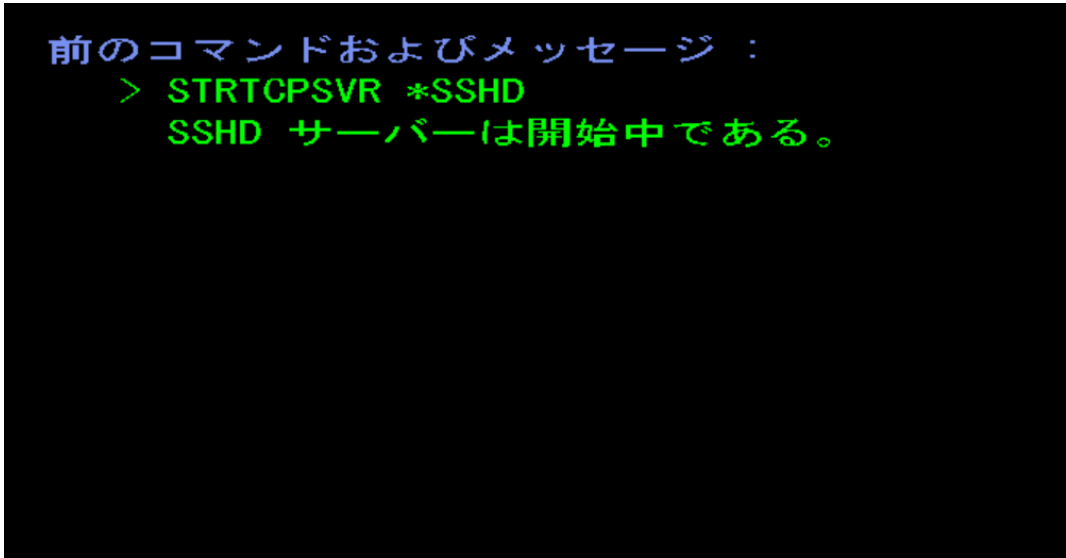
<https://github.com/ibmdb/python-ibmdb>

2. Python のインストール手順

(1) sshdの開始

5250エミュレーターで、以下のCLコマンドを実行してください。

```
STRTCPSVR *SSHD
```



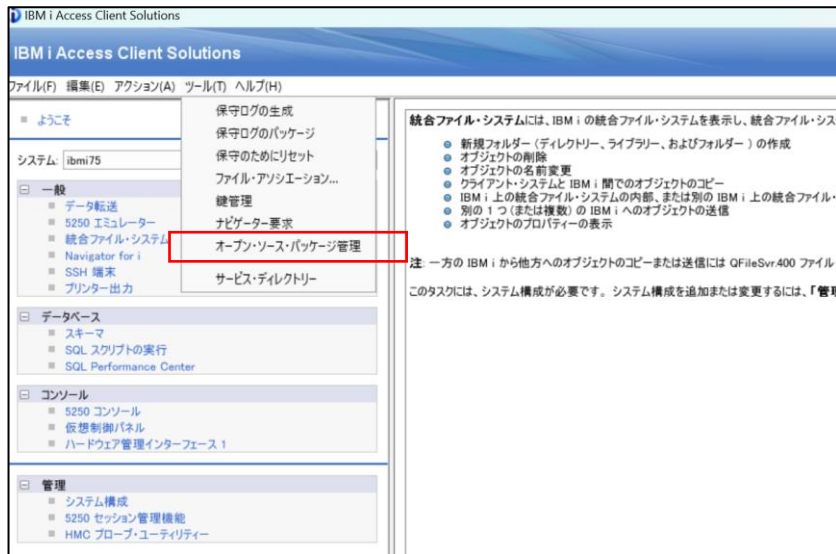
```
前のコマンドおよびメッセージ :  
> STRTCPSVR *SSHD  
SSHD サーバーは開始中である。
```


2. Python のインストール手順

(2) yumの導入-1

yumについての詳細は、「IBM iでyumを使う*」ガイドをご覧ください。
ここでは簡単な手順のみ解説します。

ACSの「オープンソースパッケージ管理」を選択



システム名、ユーザーID、パスワードを指定



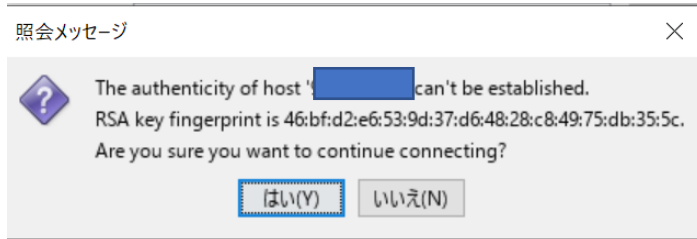
* IBM iでyumを使う：

<https://qiita.com/sugata/items/29b646a946b0c11d290b> 9

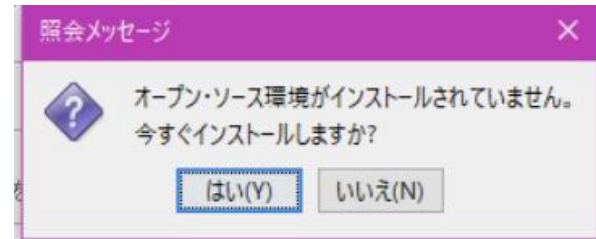
2. Python のインストール手順

(2) yumの導入-2

RSAキーの警告がでますが、「はい」を選択



下記のメッセージで「はい」を選択



2. Python のインストール手順

(2) yumの導入-3

導入が完了すると、以下のようなOSSパッケージが表示されます。

- ・基本的なパッケージと共にyumがインストールされます。

オープン・ソース・パッケージ管理

ファイル 表示 接続 ユーティリティ

接続: SAWADA@ibmi75/

インストール済みパッケージ 使用可能な更新 使用可能なパッケージ

パッケージ	バージョン	リポジトリ
libuv1	1.42.0-1	@ibmi-base
libxml2-2	2.9.4-6	@ibmi-base/7.3
libz1	1.2.12-1	@ibmi-base
ncurses-terminfo	6.0-9	@ibmi-base/7.3
nodejs18	18.5.0-1	@ibmi-base
nspr	4.32-1	@ibmi-base/7.3
nss	3.68.1-1	@ibmi-base/7.3
openssl	1.1.1q-1	@ibmi-base
openssl-devel	1.1.1q-1	@ibmi-base
openssl-engines	1.1.1q-1	@ibmi-base
p11-kit-trust	0.23.14-4	@ibmi-base/7.3
pase-includes-dummy	7.3-1	@ibmi-base/7.3
pase-libs-dummy	7.3-1	@ibmi-base/7.3
perl	5.24.1-5	@ibmi-base/7.3
python2	2.7.18-6	@ibmi-base/7.3
python2-iniparse	0.4-2	@ibmi-base/7.3
python2-pycurl	7.43.0-4	@ibmi-base/7.3
python2-rpm	4.13.1-15	@ibmi-base/7.3
python2-urlgrabber	3.102-3	@ibmi-base/7.3
python39	3.9.11-1	@ibmi-base
rpm	4.13.1-15	@ibmi-base/7.3
tn5250	0.17.5-3	@ibmi-base
unixODBC	2.3.9-1	@ibmi-base
update-alternatives	1.19.7-2	@ibmi-base/7.3
vim	8.1-3	@ibmi-base
yum	3.4.3-22	@ibmi-base/7.3
yum-metadata-parser	1.1.4-3	@ibmi-base/7.3
yum-utils	1.1.31-9	@ibmi-base/7.3

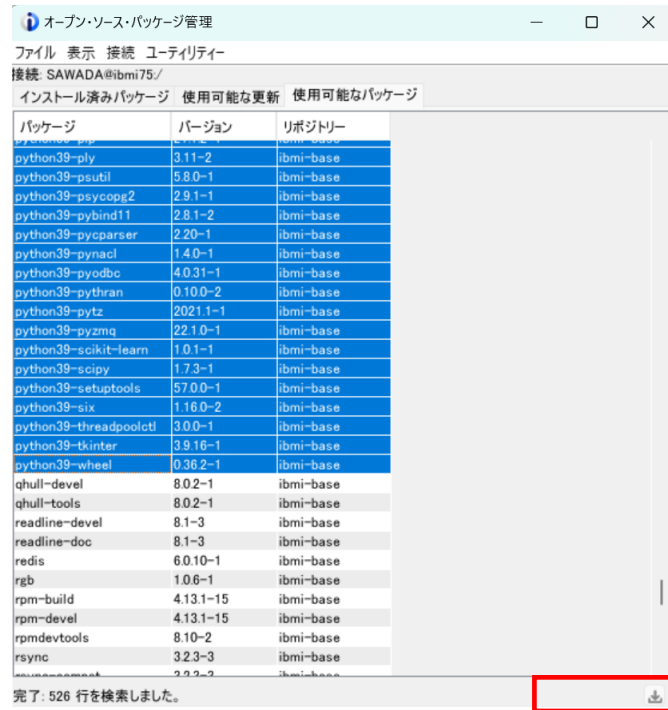
2. Python のインストール手順

(3) Pythonの導入-1

先ほど導入したyumとACSを利用してPythonを導入します。

ACSのオープンソースパッケージ管理から
「使用可能なパッケージ」タブを選択する。

Python3で始まるパッケージをすべて選択し、
「インストール」を選択する。



2. Python のインストール手順

(3) Pythonの導入-2

下記の画面が表示されるので、
「y」を選択する。

```
Package Installation
python39-threadpoolctl  noarch  3.0.0-1  ibmi-base  33 k
python39-tkinter        ppc64   3.9.16-1  ibmi-base  812 k
python39-wheel          noarch  0.36.2-1  ibmi-base  71 k
Updating:
python39                ppc64   3.9.16-1  ibmi-base  37 M
Installing for dependencies:
libX11-6                ppc64   1.7.0-3   ibmi-base  2.2 M
libX11-data             noarch  1.7.0-3   ibmi-base  375 k
libXau6                 ppc64   1.0.9-4   ibmi-base  57 k
libXdmp6                ppc64   1.1.3-4   ibmi-base  57 k
libfreetype6           ppc64   2.10.4-1  ibmi-base  1.1 M
libgfortran3           ppc64   6.3.0-29  ibmi-base  1.0 M
libgfortran5           ppc64   10.3.0-11 ibmi-base  1.4 M
libgomp1                ppc64   10.3.0-11 ibmi-base  339 k
libjpeg8                ppc64   1.5.1-2   ibmi-base  366 k
libopenblas0           ppc64   0.3.19-1  ibmi-base  7.2 M
libsodium23            ppc64   1.0.18-1  ibmi-base  515 k
libtiff5                ppc64   4.0.9-0   ibmi-base  525 k
libwebp7                ppc64   1.0.2-0   ibmi-base  1.0 M
libxcb1                 ppc64   1.13.1-2  ibmi-base  467 k
libxslt                 ppc64   1.1.29-4  ibmi-base  684 k
libyaml-0-2            ppc64   0.2.5-1   ibmi-base  128 k
libzmq5                 ppc64   4.2.5-1   ibmi-base  4.3 M
postgresql12-libpq    ppc64   12.2-2    ibmi-base  490 k
python-rpm-macros      noarch  3.6-6     ibmi-base  7.6 k
python-srpm-macros    noarch  3.6-6     ibmi-base  9.7 k
tcl                     ppc64   8.6.8-3   ibmi-base  6.0 M
tk                      ppc64   8.6.8-4   ibmi-base  3.0 M

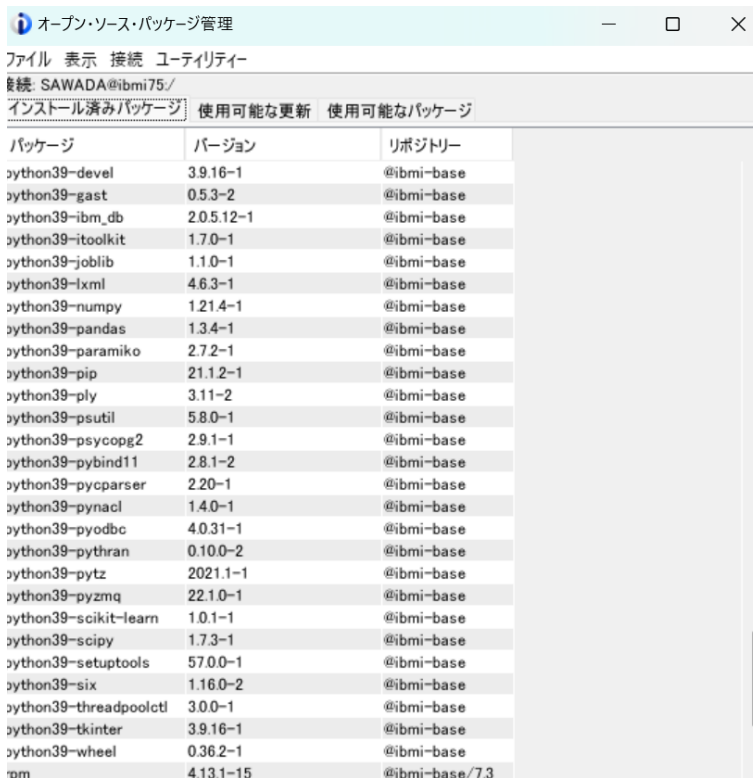
Transaction Summary
-----
Install  90 Packages
Upgrade  1 Package

Total download size: 350 M
Is this ok [y/N]:
```

2. Python のインストール手順

(3) Pythonの導入-3

Pythonのインストールが完了すると、下記のように、「インストール済みパッケージ」にPythonが表示されます。このマシンにはPython 3.9が導入済みです。



パッケージ	バージョン	リポジトリ
python39-devel	3.9.16-1	@ibmi-base
python39-gast	0.5.3-2	@ibmi-base
python39-ibm_db	2.0.5.12-1	@ibmi-base
python39-itoolkit	1.7.0-1	@ibmi-base
python39-joblib	1.1.0-1	@ibmi-base
python39-lxml	4.6.3-1	@ibmi-base
python39-numpy	1.21.4-1	@ibmi-base
python39-pandas	1.3.4-1	@ibmi-base
python39-paramiko	2.7.2-1	@ibmi-base
python39-pip	21.1.2-1	@ibmi-base
python39-ply	3.11-2	@ibmi-base
python39-psutil	5.8.0-1	@ibmi-base
python39-psycopg2	2.9.1-1	@ibmi-base
python39-pybind11	2.8.1-2	@ibmi-base
python39-pycparser	2.20-1	@ibmi-base
python39-py-nacl	1.4.0-1	@ibmi-base
python39-pyodbc	4.0.31-1	@ibmi-base
python39-pythran	0.10.0-2	@ibmi-base
python39-pytz	2021.1-1	@ibmi-base
python39-pyzmq	22.1.0-1	@ibmi-base
python39-scikit-learn	1.0.1-1	@ibmi-base
python39-scipy	1.7.3-1	@ibmi-base
python39-setuptools	57.0.0-1	@ibmi-base
python39-six	1.16.0-2	@ibmi-base
python39-threadpoolctl	3.0.0-1	@ibmi-base
python39-tkinter	3.9.16-1	@ibmi-base
python39-wheel	0.36.2-1	@ibmi-base
rpm	4.13.1-15	@ibmi-base/7.3

後の演習で利用する、ibm_db とiToolkitが導入されていることも確認してください。



パッケージ	バージョン	リポジトリ
python39-dateutil	2.8.1-2	@ibmi-base
python39-devel	3.9.16-1	@ibmi-base
python39-gast	0.5.3-2	@ibmi-base
python39-ibm_db	2.0.5.12-1	@ibmi-base
python39-itoolkit	1.7.0-1	@ibmi-base
python39-joblib	1.1.0-1	@ibmi-base
python39-lxml	4.6.3-1	@ibmi-base
python39-numpy	1.21.4-1	@ibmi-base
python39-pandas	1.3.4-1	@ibmi-base
python39-paramiko	2.7.2-1	@ibmi-base
python39-pip	21.1.2-1	@ibmi-base

2. Python のインストール手順

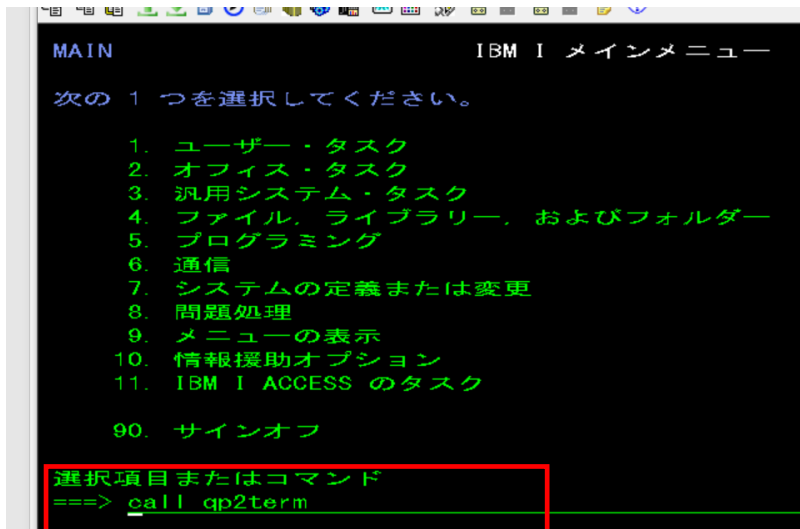
(3) Pythonの導入確認 - 1

Pythonは、他の多くのオープンソーステクノロジーと同様に、IBM iのPASE環境で実行されます。

Pythonが正しくインストールされていることを確認してみましょう。5250は、英語小文字が表示できる設定（ホストコードページ939など）にしておきます。

5250画面から、下記のコマンドでPASE環境に入る。（「call qp2term」コマンド）

下記の画面になります。



A screenshot of the IBM i MAIN menu. The text is as follows:

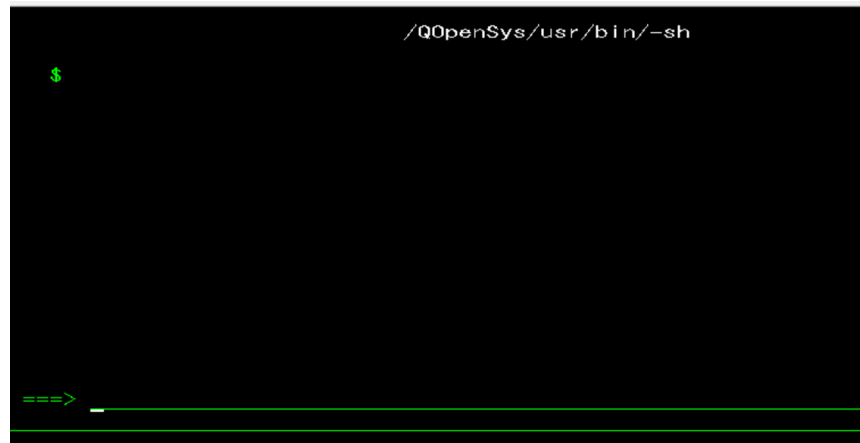
```
MAIN                               IBM I   メインメニュー

次の 1 つを選択してください。

1. ユーザー・タスク
2. オフィス・タスク
3. 汎用システム・タスク
4. ファイル、ライブラリー、およびフォルダー
5. プログラミング
6. 通信
7. システムの定義または変更
8. 問題処理
9. メニューの表示
10. 情報援助オプション
11. IBM I ACCESS のタスク

90. サインオフ

選択項目またはコマンド
==> call qp2term
```



A screenshot of the PASE shell prompt. The text is as follows:

```
/QOpenSys/usr/bin/-sh

$

==> _
```

2. Python のインストール手順

(3) Pythonの導入確認 - 2

yumでインストールされたOSSは、IFS上の /Opensys/pkgs/binの下に導入されますので、環境変数PATHを変更しておきます。

下記のコマンドを使用してPATHを変更します

```
export PATH=/QOpenSys/pkgs/bin:$PATH
```

「python3」コマンドで、導入されているPythonのバージョン (v3.9.16) が確認できました。
これで導入確認は完了です。

```
$  
> export PATH=/QOpenSys/pkgs/bin:$PATH
```



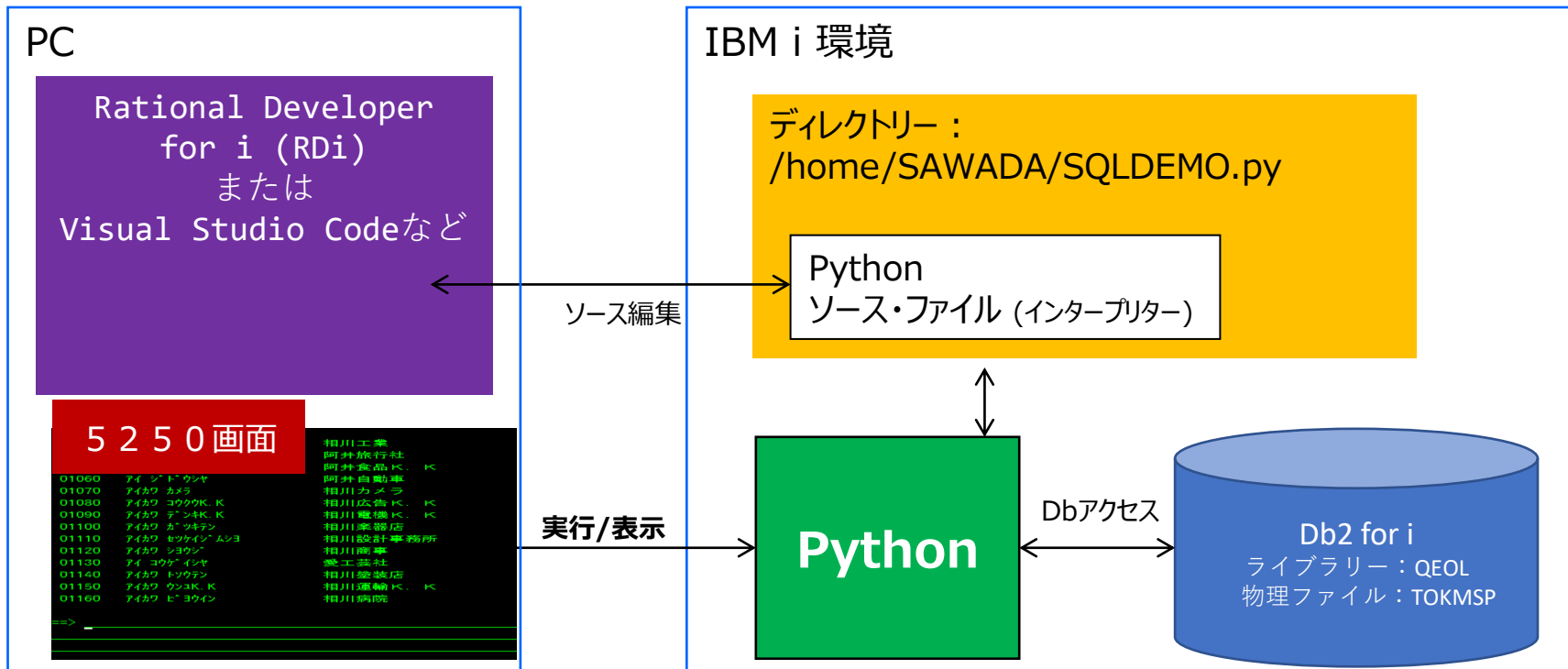
```
> python3  
Python 3.9.16 (main, Feb 1 2023, 13:01:20)  
[GCC 6.3.0] on os400  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```


3. Pythonで簡単なプログラムを作ってみよう

1) 既存のデータベースにアクセスしてみよう-1 (実行環境イメージ)

Python のソース編集 を 使い慣れたエディターで行います

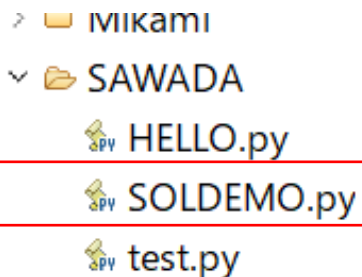
ここではRSE を使用して、IBM i の IFS のソースファイルを直接編集します。



3. Pythonで簡単なプログラムを作ってみよう

2) 既存のデータベースにアクセスしてみよう-2 (ソースコードの作成)

RDide、IFS上の/home/SAWADAフォルダーに、「SQLDEMO.py」を作成します。



```
>  📁 MIKAMI
  >  📁 SAWADA
     📄 HELLO.py
     📄 SQLDEMO.py
     📄 test.py
```

下記のように入力し、保存してください。

```
import ibm_db_dbi as db2
Conn
=db2.connect("DATABASE=*LOCAL", "UID=SAWADA",
"PWD=SAWADA")
cursor = conn.cursor()
sql = "SELECT * FROM QEOL.TOKMSP"
cursor.execute(sql)
for r in cursor.fetchall():
    print(r[0],end=' ')
    print(r[1],end=' ')
    print(r[2])
```

3. Pythonで簡単なプログラムを作ってみよう

2) 既存のデータベースにアクセスしてみよう-3 (プログラムの実行)

PASE環境 (call qp2term) で、Pythonのプログラムを実行します。

スクロールすると、QEOLライブラリーのTOKMSPのファイルが表示されます。

```
> python SOLDEMO.py
01010 アイリョカン
01020 アイコウキョウ
```

阿井旅館
阿井工業



```
01030 アイワ コウキョウ 相川工業
01040 アイ リョコウシャ 阿井旅行社
01050 アイ ショクトウK. K 阿井食品K. K
01060 アイ シフトウシャ 阿井自動車
01070 アイワ カメラ 相川カメラ
01080 アイワ コウクウK. K 相川広告K. K
01090 アイワ テンキK. K 相川電機K. K
01100 アイワ カツキテン 相川楽器店
01110 アイワ セツケイシムシヨ 相川設計事務所
01120 アイワ ショウジ 相川商事
01130 アイ コウゲイシヤ 愛工業社
01140 アイワ トソウテン 相川塗装店
01150 アイワ ウンユK. K 相川運輸K. K
01160 アイワ ヒョウイン 相川病院
```

```
==> _
```

3. Pythonで簡単なプログラムを作ってみよう

2) 既存のデータベースにアクセスしてみよう-4 (SQLDEMOのプログラム解説)

簡単にプログラムを解説します。

```
import ibm_db_dbi as db2
```

IBM iデータベースアクセスのためのモジュールを使うのに必要な設定

```
conn  
=db2.connect("DATABASE=*LOCAL", "UID=SAWADA", "PWD=  
SAWADA")
```

IBM iのデータベースを指定してユーザーID/PASSWORDを入力

```
cursor = conn.cursor()  
sql = "SELECT * FROM QEOL.TOKMSP"  
cursor.execute(sql)
```

カーソル作成とSQL実行

```
for r in cursor.fetchall():  
    print(r[0],end='    ')  
    print(r[1],end='    ')  
    print(r[2])
```

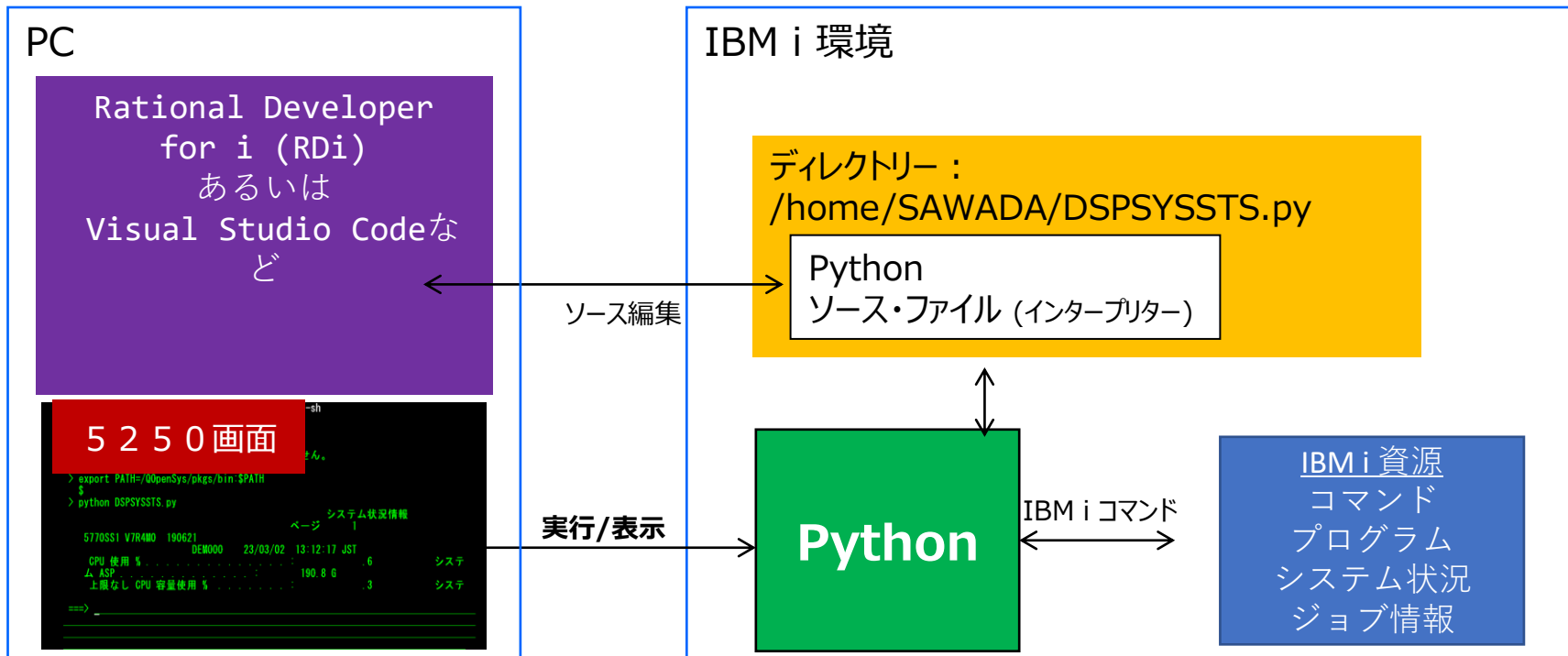
レコードにフェッチし、1件ごとに出カし表示します

3. Pythonで簡単なプログラムを作ってみよう

3) itoolkitを使ってIBM iのコマンドにアクセスしてみよう-1 (実行環境イメージ)

Python のソース編集 を 使い慣れたエディターで行います

RSE を使用して、IBM i の IFS のソースファイルを直接編集します.他の使い慣れたエディターでもOKです。

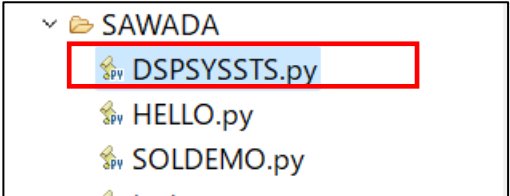


3. Pythonで簡単なプログラムを作ってみよう

3) itoolkitを使ってIBM iコマンドにアクセスしてみよう-2 (ソースコード入力)

下記のように入力し、保存してください。

(2)と同様に、RDiで、IFS上の
/home/SAWADAフォルダーに、
DSPSYSSTS.pyを作成します。



SAWADA
DSPSYSSTS.py
HELLO.py
SOLDEMO.py



```
import ibm_db_dbi as db2
from itoolkit import *
from itoolkit.transport import DatabaseTransport

conn = db2.connect("DATABASE=*LOCAL", "UID=SAWADA", "PWD=SAWADA")
itool = iToolKit()
itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))

itransport = DatabaseTransport(conn,ctl="*pase(1208/5035)")
# xmlservice
itool.call(itransport)

# output
dspsyssts = itool.dict_out('dspsyssts')
if 'error' in dspsyssts:
    print (dspsyssts['error'])
    exit()
else:
    print (dspsyssts['dspsyssts'])
```

3. Pythonで簡単なプログラムを作ってみよう

3) itoolkitを使ってIBM iコマンドにアクセスしてみよう-3 (プログラム実行)

下記のようにDSPSYSSTS (システム状況情報) が表示されます。

PASE環境 (call qp2term) で、Pythonのプログラムを実行します。

```
> export PATH=/Q0openSys/pkg/bin:$PATH
$
> python DSPSYSSTS.py
```



```

                                     システム状況情報
                                     ページ 1
5770SS1 V7R4M0 190621
      DEM000 23/03/02 13:28:14 JST
CPU 使用 % . . . . . 7 システ
ム ASP 190.8 G . . . . . 3 システ
上限なし GPU 容量使用 % . . . . .
ム ASP 使用 % . . . . . 81.4616
経過時間 . . . . . 00:00:01 合計補
助記憶域 . . . . . 190.8 G
システム内のジョブ . . . . . 451 現行使
用一時 24523 M . . . . . ピーク
永久アドレス % . . . . . 0.10
使用一時 24523 M . . . . .
一時アドレス % . . . . . 0.12

                                     ページ
システム プール 予約済 MAX -----DB----- 非 DB----- ACT-
> WAIT-> ACT->
プール サイズ M サイズ M ACT 不在 送り 不在 ページ WAI
T INEL INEL プール サブシステム オプション
8 1 1314.12 677.08 +++++ 0 0 0 0 107.
0 0 *MACHINE *FIXED
5 2 12454.50 10.39 195 1.7 1.7 8 3.5 5552.
0 0 *BASE *CALC
0 3 162.37 00 5 0 0 0 0
0 0 *SPOOL *FIXED
0 4 2306.98 00 406 0 0 0 0
0 0 *INTERACT *FIXED
                                     ***** リストの終わ
り *****

```

3. Pythonで簡単なプログラムを作ってみよう

3) itoolkitを使ってIBM iコマンドにアクセスしてみよう-4 (プログラムの解説)

```
import ibm_db_dbi as db2
from itoolkit import *
from itoolkit.transport import DatabaseTransport
```

IBM i資源にアクセスするための
itoolkitを使用するのに必要な設
定です。

```
conn = db2.connect("DATABASE=*LOCAL", "UID=SAWADA", "PWD=SAWADA")
itool = iToolkit()
```

```
itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))
```

IBM iとの接続(IP,ユーザー
IDなどを設定します

```
itransport = DatabaseTransport(conn,ctl="*pase(1208/5035)")
```

```
# xmlservice
```

```
itool.call(itransport)
```

DSPSYSSTSのコマンドを
定義しています。

```
# output
```

```
dspsyssts = itool.dict_out('dspsyssts')
```

```
if 'error' in dspsyssts:
```

```
    print (dspsyssts['error'])
```

```
    exit()
```

```
else:
```

```
    print (dspsyssts['dspsyssts'])
```

結果の表示をします

4. PythonでWebフレームワークを使ってみよう



1) シンプルなWebフレームワーク「Bottle」のご紹介

- Bottle は、Python用の高速でシンプルかつ軽量な WSGIマイクロWebフレームワーク*です。単一のファイルモジュールとして配布され、Python標準ライブラリー以外の依存関係はありません。
- 最低限のPythonの知識で始めることができます。コーディング初心者でも理解し易いです。
- 現在（2023/03）の安定版は、v0.12です。IBM iからも、PIPでダウンロードできます。

*WSGI (**Web Server Gateway Interface**) は、Pythonにおける、WebアプリケーションとWebサーバを接続する標準仕様を定めるものである。これによって、WSGIに対応したWebアプリケーション（やフレームワーク）は、WSGIに対応した任意のWebサーバ上で運用できるようになる。つまり、アプリケーション側がWSGIに対応していれば、アプリケーションのコードに修正を加えることなく、WSGI対応サーバを自由に選択することができ、高い可搬性が得られる。（出典: フリー百科事典：ウィキペディア (Wikipedia)

4. PythonでWebフレームワークを使ってみよう

2) Bottleの導入と稼働テスト

- 下記のコマンドで導入できます。5250画面から、Call qp2termを使って、pipコマンドを入力してください。
- PIPコマンドは、pythonに標準で組み込まれているパッケージ管理用システムで、python3系はpip3コマンドです。

```
> export PATH=/QOpenSys/pkgsrc/bin:$PATH
$
> pip3 install bottle
```

- 下記のようになればokです。

```
> pip3 install bottle
Collecting bottle
  Downloading bottle-0.12.24-py3-none-any.whl (90 kB)
    [K | ..... |
90 kB 341 kB/s
[?25hInstalling collected packages: bottle
Successfully installed bottle-0.12.24
[33mWARNING: Running pip as root will break packages and permissions. You should install packages reliably by using venv: https://pip.pypa.io/warnings/venv
nv [0m
```

- Bottleのバージョンは、0.12.24が導入されました。Pip3 listコマンドで導入されたパッケージを確認できます。

```
> pip3 list
Package            Version
-----
asn1crypto         0.24.0
bcrypt             3.1.4
bottle             0.12.24
cffi               1.11.5
cryptography       2.8
ibm-db             2.0.5.12
idna               2.8
itoolkit           1.6.1
Jinja2             2.11.2
lxml               4.2.1
```

4. PythonでWebフレームワークを使ってみよう

2) Bottleの導入と稼働テスト

まずは、簡単なプログラムで稼働テストをしてみましょう。

- サンプルプログラムがBottleの公式ホームページ (<https://bottlepy.org/docs/dev/>) にあります。これを稼働してみましょう。
- 下記のプログラム(hello.py)を作成して、IBM iの/home/SAWADA に送ります。

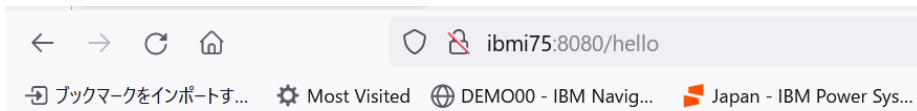
```
from bottle import route, run
@route('/hello')
def hello():
    return "Hello World!"

run(host='ibmi75', port=8080, debug=True)
```

- 5250画面から、下記のように起動します。

```
> python hello.py
Bottle v0.12.24 server starting up (using WSGIRefServer())..
Listening on http://ibmi75:8080/
Hit Ctrl-C to quit.
```

- ブラウザー画面で下記のように入力 (<http://hostname:8080/hello>) し、「Hello World」が表示されたらOKです。



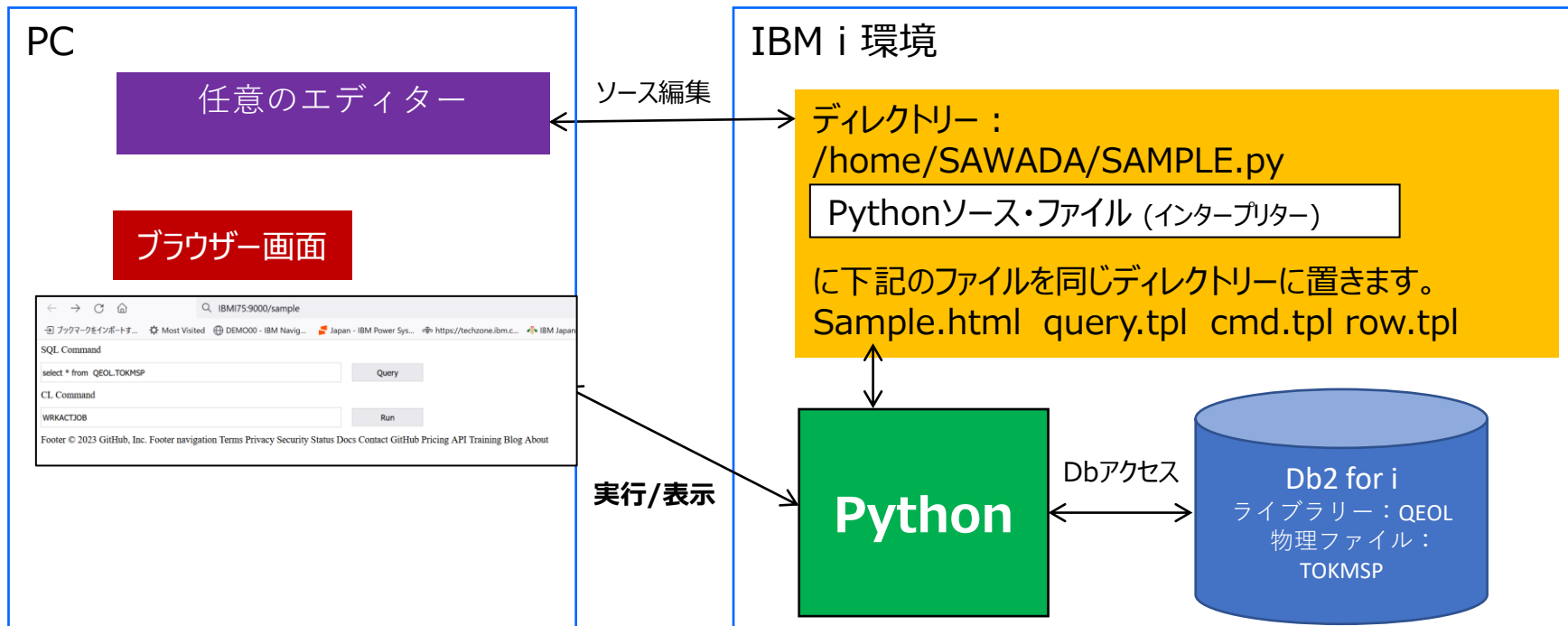
4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-1 (実行環境イメージ)

次に、3で活用した、IBM iのDb2のアクセスと、CLコマンドにWebからアクセスする簡単なプログラムを作ってみましょう。

- ・このサンプルプログラムは、下記のコードを基に作成しました。

<https://github.com/IBM/ibmi-oss-examples/tree/master/python/bottle-example>



4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-2 (ソースコード1)

(2)と同様にIFS上の/home/SAWADA
フォルダーに、下記の5つの
ファイルを置きます。まずは本体のファ
イル (SAMPLE.py)です。

/home/SAWADA		
アイコン	名前	サイズ
	cmd.tpl	
	DSPSYSSTS.py	
	HELLO.py	
	query.tpl	
	row.tpl	
	sample.html	
	SAMPLE.py	
	SOLDEMO.py	



```

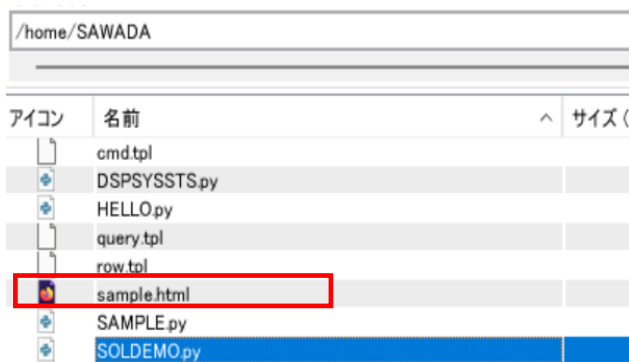
from bottle import request, get, post, static_file, route, run, template
import ibm_db_dbi as dbi
from itoolkit import *
from itoolkit.db2.idb2call import * #for local jobs
version = tuple(map(int, dbi.__version__.split('.')))
if version < (2, 0, 5, 5):
    raise Exception("Need ibm_db_dbi 2.0.5.5 or higher to run, you have " + dbi.__version__)
@route('/sample')
def sample():
    return static_file('sample.html', root='.')
@route('/query', method='POST')
def query_ibm_db():
    statement = request.forms.get('sql')
    conn = dbi.connect("DATABASE=*LOCAL", "UID=SAWADA", "PWD=SAWADA")
    cur = conn.cursor()
    cur.execute(statement)
    headers = [descr[0] for descr in cur.description]
    return template('query', headers=headers, rows=cur)
@route('/cmd', method='POST')
def cmd_toolkit():
    cl_statement = request.forms.get('cl')
    # xmlservice
    conn = dbi.connect("DATABASE=*LOCAL", "UID=SAWADA", "PWD=SAWADA")
    itool = iToolkit()
    itransport = iDB2Call(conn)
    itool.add(iCmd5250(cl_statement, cl_statement))
    itransport = DatabaseTransport(conn,ctl="*pase(1208/5035)")
    itool.call(itransport)
    # results from list
    data = ''
    for output_outer in itool.list_out():
        for output_inner in output_outer:
            data += output_inner
    return template('cmd', data=data)
run(host='DEMO', port=9000, debug=True, reloader=True)

```

4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-3 (ソースコード2)

次に、本体のHTMLファイル
(SAMPLE.HTML)です。

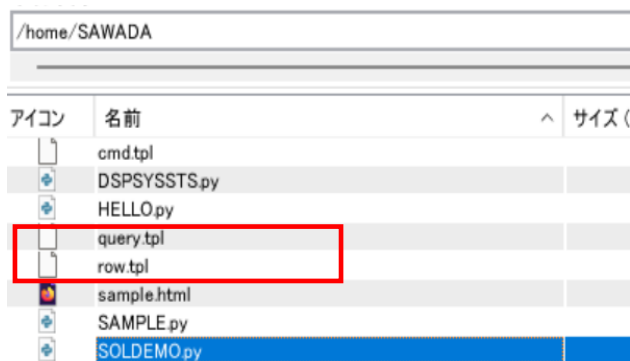


```
<!DOCTYPE HTML>
<html lang="en-US">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Python for i Sample</title>
</head>
<style>
input
{
  height:30px;
  border:#ccc solid 1px
}
input[type="text"]
{
  width:500px
}
input[type="submit"]
{
  margin:1em;
  width:120px
}
</style>
</head>
<body>
  <form name="input" action="query" method="post">
    <div>SQL Command </div>
    <input type="text" name="sql" placeholder="SELECT * FROM ..."/>
    <input type="submit" value="Query"/>
  </form>
  <form name="input" action="cmd" method="post">
    <div>CL Command </div>
    <input type="text" name="c1" placeholder="WRKSYSSTS"/>
    <input type="submit" value="Run"/>
  </form>
</body>
</html>
```

4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-4 (ソースコード3)

次に、SQL検索用のHTMLファイル
(query.tpl)です。



queryから呼び出している
HTMLファイル (row.tpl)です。

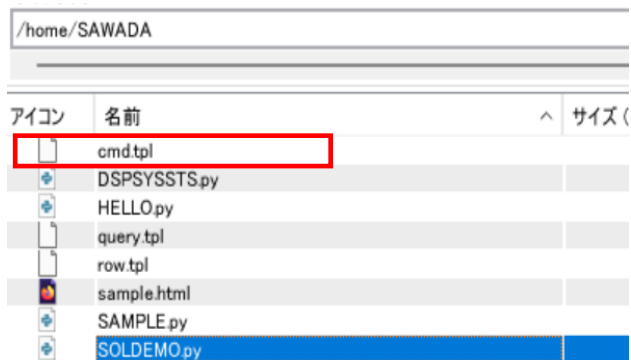
```
<tr>
% for value in values:
    % if header:
        <th>{{value}}</th>
    % else:
        <td>{{value}}</td>
    % end
% end
</tr>
```

```
<html>
<head>
<style>
table, th, td
{
    border: 1px solid black;
    border-collapse: collapse;
}
th, td
{
    padding: 5px;
    text-align: left;
    width: 200px;
}
tr:nth-child(even)
{
    background-color: #f3f3f3;
}
tr:nth-child(odd)
{
    background-color: #fff;
}
th
{
    background-color: #008ABF;
    color: white
}
</style>
<head>
<body>
<table>
% include('row', values=headers, header=True)
% for row in rows:
    % include('row', values=row, header=False)
% end
</table>
</body>
</html>
```

4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-5 (ソースコード4)

次に、コマンド用のHTMLファイル (cmd.tpl)です。



アイコン	名前	サイズ
	cmd.tpl	
	DSPSYSSTS.py	
	HELLO.py	
	query.tpl	
	row.tpl	
	sample.html	
	SAMPLE.py	
	SOLDEMO.py	

```
<!DOCTYPE HTML>
<html lang="en-US">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Python for i Sample</title>
  </head>
  <body>
    <p>
      <pre>{{data}}</pre>
    </p>
  </body>
</html>
```


4. PythonでWebフレームワークを使ってみよう

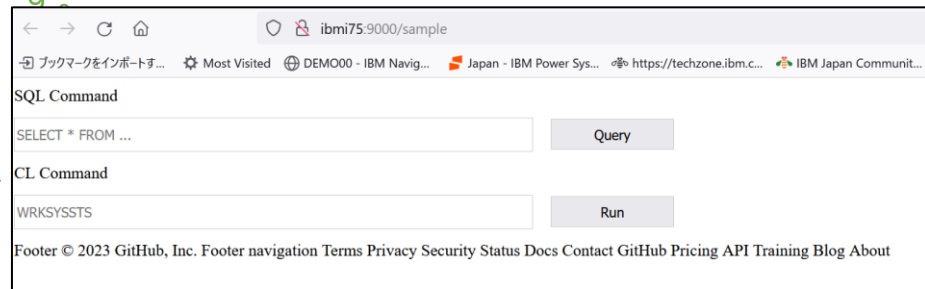
3) BottleのWebフレームワークを利用したWebアプリ-6 (プログラム実行1)

(2)と同様に、PASE環境 (call qp2term) で、Pythonのプログラム(SAMPLE.py)を実行します。ポート9000で待ち受けになります。

```
> export PATH=/QOpenSys/pkgsrc/bin:$PATH
$
> python SAMPLE.py
Bottle v0.12.24 server starting up (using WSGIRefServer()).
Listening on http://ibmi75:9000/
Hit Ctrl-C to quit.
```



下記のようにブラウザから
<http://hostname:9000/sample> を実行しま
す。



4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-7 (プログラム実行2)

SQL Commandへ下記のように
Select * from QEOL.TOKMSP 入力し、Queryボタン
を実行します。

SQL Command	
<code>select * from QEOL.TOKMSP</code>	Query
CL Command	
WRKSYSSTS	Run
Footer © 2023 GitHub, Inc. Footer navigation Terms Privacy Security Status Docs Contact GitHub P	



下記のように表形式で、得意先マスターのデータが表示されます。
select文を変えて、自社のデータを表示してみてください。

TKBANG	TKNAKN	TKNAKJ	TKADR1	TKADR2	TKTIKU	TKPOST	TKTELE	TKGURI	TKNURI	TKZURI	TKUZAN	TKGEND	TKNYUK	TKSIME
01010	阿井旅館	東京都渋谷区	桜ヶ丘30	02	150	03-504-9293	698500	4086300	6615600	1000000	1100000	880427	2	
01020	阿井工業	東京都渋谷区	渋谷1-3	02	150	03-535-5951	452800	2713600	4483700	670000	1150000	880607	2	
01030	相川工業	東京都世田谷区	若林4-24	06	154	03-964-6406	136200	243000	796600	110000	1120000	880619	1	
01040	阿井旅行社	東京都品川区	東五反田1-11	01	140	03-934-7946	3028300	18083100	27670700	9000000	1200000	880402	4	
01050	阿井食品K.K	東京都荒川区	荒川5-42	17	116	03-360-6701	541600	2697600	4279600	1300000	1350000	880121	1	
01060	阿井自動車	東京都港区	芝公園1-2	14	105	03-860-2932	541100	2819100	4599900	1400000	1250000	880525	5	
01070	相川カメラ	東京都新宿区	四谷1-16	03	160	03-354-3018	367900	2011700	3914500	960000	1000000	880611	1	
01080	相川広告K.K	東京都渋谷区	広尾3-9	02	150	03-368-6366	318000	1461700	2205800	290000	2000000	880214	2	
01090	相川電機K.K	東京都北区	滝の川7-17	14	114	03-749-6271	877100	4731400	7135500	940000	900000	880519	2	
01100	相川楽器店	東京都港区	虎ノ門3-21	22	105	03-922-1801	501300	2973300	4800100	590000	1300000	880615	6	

4. PythonでWebフレームワークを使ってみよう

3) BottleのWebフレームワークを利用したWebアプリ-8 (プログラム実行3)

CL Commandへ下記のように
WRKSYSSTS (システム状況情報) を入力し、
Runボタンを実行します。

SQL Command

SELECT * FROM ... Query

CL Command

WRKSYSSTS Run

Footer © 2023 GitHub, Inc. Footer navigation Terms Privacy Security Status Docs Contact GitHub



下記のようにWRKSYSSTSのデータが表示されます。
その他のCLコマンドも試してみてください。

```

6770SS1 V7R4M0 190821          システム状況情報          DEM000  23/03/03  14:51:22 JST
CPU使用%                   : 1.0          システムASP                   : 190.8 G
上層なしCPU容量使用%      : 5           システムASP使用%              : 80.870%
経過時間                   : 00:00:01   合計補助記憶域                : 190.8 G
システム内のジョブ        : 513          空き使用一時                  : 24608 M
永久アドレス%             : .010       ログ使用一時                  : 24635 M
一時アドレス%             : .012

```

ジョブ ジョブ ジョブ	ジョブ ジョブ ジョブ	予約済 予約済 予約済	MAX MAX MAX	ACT ACT ACT	DB DB DB	非DB 非DB 非DB	ACT-> ACT-> ACT->	WAIT-> WAIT-> WAIT->	INEL INEL INEL	ACT-> ACT-> ACT->	ジョブ ジョブ ジョブ	ジョブ ジョブ ジョブ	ジョブ ジョブ ジョブ
1	1314.14	677.09	++++	.0	.0	.0	.0	110.7	.0	.0	*MACHINE	ジョブ	*FIXED
2	11240.33	10.39	195	.0	.0	.0	.0	5479.7	.0	.0	*BASE	ジョブ	*CALC
3	162.37	.00	5	.0	.0	.0	.0	.0	.0	.0	*SPOOL	ジョブ	*FIXED
4	3521.14	<.01	406	.0	.0	.0	.0	.0	.0	.0	*INTERACT	ジョブ	*FIXED

***** リストの終わり *****

4. 補足情報

(1) プログラミング言語 Pythonの紹介

<https://www.python.jp/pages/about.html>

(2) IBM i で Python を使ってみよう

<http://www63.tok2.com/home2/jd4/SetupPython400.html>

(3) IBM iにPythonの実行環境をつくる

<https://qiita.com/sugata/items/dd96bb5c2b8a2f0efd1f>

(4) IBM i OSS DocsのPython Usage notes

<https://ibmi-oss-docs.readthedocs.io/en/latest/python/README.html>

(5) itoolkitのドキュメント

https://python-itoolkit.readthedocs.io/_/downloads/en/latest/pdf/

(6) Webフレームワーク bottleの公式サイト

<https://bottlepy.org/docs/dev/>

IBM Power Salonのご案内



IBM Powerユーザーのための自由な語り場がオープンします。
オンラインサロンで、お客様同士、IBMスペシャリストと繋がりませんか？
是非、お気軽にご参加ください。

日時：第17回 2023年 4月12日(水) 9:00-10:00

内容：お客様によるDX事例、クラウド活用事例
IBM i/AIX/Linuxの技術情報、サポート情報
IBMスペシャリストによるQ&A 他



第16回	3月8日(水)	キョーラク株式会社様 AI実装プロジェクト プロジェクトメンバー講演 済み
第17回	4月12日(水)	光世証券株式会社様 ご講演予定
第18回	5月10日(水)	株式会社 フェリシモ様 ご講演予定
第19回	6月14日(水)	お客様講演予定
第20回	7月12日(水)	お客様講演予定
	8月	夏季休暇のため、休会

主催：日本アイ・ビー・エム（株）IBM Power 事業部

参加方法：オンライン開催
どなたでも参加可能、無料
事前申し込み不要

ご参加URL：<https://ibm.biz/Powersalon-webex>

お問い合わせ：NO1POWER@jp.ibm.com

IBM i 情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i World 2021 オンデマンド・セミナー

<https://ibm.biz/iworld2021>

IBM i ホワイトペーパー 2021年日本語版

<https://www.ibm.com/downloads/cas/JB8AX09V>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/>

IBM i 情報 Facebook

<https://www.facebook.com/iusersjapan>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseriess.html>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)
「IBM i」「9009-41G」などPTF情報の必要な製品を
選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 7.4 技術資料

<https://www.ibm.com/docs/ja/i/7.4>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。